

We Claim:

1. A method of reordering packet fragments, said packet fragments being derived from a packet which has been divided into at least two fragments, each of said at least two fragments having a sequence number denoting a position in a proper sequence for correct ordering of each fragment for a reassembly of said packet, each fragment being represented by a head pointer, the method comprising:

a) receiving a received head pointer for a received fragment;

b) determining a sequence number for said received fragment;

c) determining a first slot position for said received head pointer in a first tier pointer array, said first slot position being determined by the sequence number for said received fragment, said first slot position having a tier pointer pointing to a second tier pointer array;

d) determining a second slot position for said received head pointer in the second tier pointer array, said second slot position being determined by the sequence number for said received fragments;

e) placing said head pointer in said second slot position; and

f) repeating steps a) - e) until a sequence of received head pointers stored in said second tier array indicates a complete sequence,

wherein

a complete sequence is a sequence of head pointers that does not have a gap and includes a fragment that is an end fragment for said packet.

2. A method of reordering packet fragments, said packet fragments being derived from a packet which has been divided into at least two fragments, each fragment having a sequence number denoting a position in a proper sequence for correct ordering of each fragment for a reassembly of said packet, each fragment being represented by a head pointer, the method comprising:

a) receiving a received head pointer for a received fragment;

b) determining a sequence number for said received fragment;

c) determining a slot position for said received head pointer in a pointer array, said slot position being determined by the sequence number for said received fragments;

d) placing said head pointer in said slot position;

e) repeating steps a) -d) until a sequence of received head pointers stored in said array indicates a complete sequence,

wherein a complete sequence is a sequence of head pointers that does not have a gap and includes a fragment that is an end fragment for said packet.

3. A method as in claim 2 wherein said pointer array is found by referring to a locator pointer entry in a locator pointer array, said pointer entry being determined by said sequence number for said received fragment.

4. A method of resequencing fragments forming portions of a data packet into a proper sequence for a reassembly of said packet, each fragment having a

sequence number denoting a proper placement of said fragment in said reassembly, the method comprising:

a) receiving a head pointer for a received fragment, said head pointer representing said received fragment;

b) determining a reassembly session number for said received fragment, said reassembly session number denoting a home packet of which said received fragment is a part;

c) determining if said home packet of which said received fragment is a part is currently being reassembled;

d) if said home packet is currently being reassembled, determining which pointer array is being used in a reassembly of said home packet and placing said head pointer in said pointer array;

e) if said home packet is not currently being reassembled, initiating a reassembly of said home packet by allocating a pointer array for said reassembly of said home packet;

f) determining a slot position for said head pointer in said pointer array, said slot position being determined by the number for said received fragment;

g) placing said head pointer in said slot position; and

h) repeating steps a) - g) until a sequence of received head pointers stored in said array indicates a complete sequence,

wherein a complete sequence is a sequence of head pointers that does not have a gap and includes a fragment that is an end fragment for said packet.

5. A method as in claim 4 wherein said pointer array is found by referring to a locator pointer entry in a locator pointer entry in a locator pointer array, said pointer entry being determined by said sequence number for said received fragment.

6. A method of processing data frames in a multiple channel system, said data frames containing a payload, said payload comprising portions of a larger data transmission unit, the method comprising:

- a) receiving a data frame;
- b) determining with which channel said data frame is associated;
- c) extracting a payload of said data frame;
- d) storing said payload at a memory location specifically associated with said channel;
- e) if previous payloads already stored at said memory location, appending said payload to said previous payloads; and
- f) repeating steps a) - e) until a specific condition is met, said specific condition being chosen from the group comprising:
 - an amount of data stored in said memory location reaches a predetermined value; and
 - a payload received and data stored in said memory location relate to different data transmission units.

7. A method as is claim 6 further including reading out said data in said memory locations and transmitting said data to a different processing unit when said specific condition is met.

8. A method as in claim 6 wherein said data transmission unit is a data packet used for transferring data in network.

9. A method as in claim 6 wherein said data transmission unit is a fragment containing a portion of a data packet used in transferring data in a network.

10. A method as in claim 6 wherein said memory location is within a static random access memory (SRAM) bank.

11. A method as in claim 1 wherein said head pointer is a beginning pointer in a linked list of pointers, each pointer in said linked list of pointers pointing to a memory location in a random access memory (RAM) bank.

12. A method as in claim 2 wherein said head pointer is a beginning pointer in a linked list of pointers, each pointer in said linked list of pointers pointing to a memory location in a random access memory (RAM) bank.

13. A method of rebuilding fragments each of said fragments being a data transmission unit having as payload a portion of data packet, said method comprising:

- a) receiving a first chunk of data comprising a first segment of a fragment;
- b) storing said chunk of data in a first block of memory;

c) receiving a second chunk of data comprising a second segment of said fragment;

d) placing said second chunk of data in a second block of memory contiguous to said first block of memory;

e) linking said second contiguous block of memory with said first contiguous block of memory; and

f) repeating steps a) - e) until said fragment is complete,

wherein

said first chunk and said second chunk are received in a sequence which preserves a data integrity of said fragment.

14. A method as in claim 13 wherein said first and second blocks of memory are located in a random access memory (RAM) bank.

15. A fragment for use in transmitting data between two nodes in a network, said fragments having:

- payload comprising a portion of a data packet;
- a sequence number denoting a proper placement of said fragment in a sequence of fragments when said data packet is being reassembled, and
- a reassembly session number identifying which data packet said fragment is a part of.

16. A fragment as in claim 15 further including an end bit which is set when said fragment is a final fragment in said sequence.

17. A packet for use in transmitting data between two nodes in a network, and for use in a multiple link data transmission system, said packet having a connection identifier denoting to which group of links said packet belongs.

18. A system for rebuilding fragments, each of said fragments being a data transmission unit having as a payload a portion of data packet, the system comprising:

- a memory bank for storing data chunks, each data chunk being a portion of a fragment;

- a linked list pointer table, said pointer table having a record of which specific memory locations in said memory bank are used by specific data chunks wherein

each data chunk relating to a specific fragment is stored in a specific memory location, an address of such specific memory location being placed in said pointer table and linked to addresses of previously stored data chunks.

19. A system for resequencing fragments forming portions of a data packet, each of said fragments having a sequence number denoting a proper placement of said fragment in said reassembly and each of said fragments having a packet session number denoting a home packet of which said fragment is a part of, the system comprising:

- at least one pointer array for each resequencing operation for storing head pointers, each head pointer representing a received fragment, each head pointer being associated with a slot in a pointer array

based on the sequence number of a fragment represented by said head pointer; and

- a lookup engine for determining if a fragment represented by a head pointer has a reassembly session number matching a resequencing operation being performed,

wherein

in the event a fragment has a reassembly session number matching a resequencing operation being performed, its head pointer is placed in a pointer array for said resequencing operation.

20. A system as in claim 19 further including at least one locator/pointer array, said at least one locator/pointer array referencing said at least one pointer array such that each entry in a locator/pointer array refers to a pointer array.

21. A system for processing data frames in a multiple channel data transmission system, said data frames containing a payload, said payload comprising portions of a larger data transmission unit, the system comprising:

- a data frame engine for extracting a payload from said data frames;

- a partial packet processor for storing payloads extracted from said data frames;

- an internal memory bank controlled by said processor such that said payloads are stored in said memory bank,

wherein

a specific memory location in said memory bank is allocated for each channel such that payloads from a specific channel are stored in said memory location.

22. A system as in claim 21 wherein if a received payload from a specific channel and previously received payloads stored in said specific memory location allocated said specific channel are from a single larger data unit, said received payload is appended to said previously received payloads.

23. A system as in claim 21 wherein if a received payload from a specific channel and previously received payloads stored in said specific memory location allocated for said specific channel are not from a single larger data unit, said previously received payloads are retrieved from said bank and transferred to a next stage and said received payload is stored in said specific memory location.

24. A method of selecting a link on which to transmit data in a multiple link system, the method comprising:

- a) determining an amount of data queued for transmission on each link in said multiple link system;
- b) determining which link in said multiple link system has the most data queued for transmission;
- c) selecting a link with a least amount of data queued for transmission as the link on which to transmit data; and
- d) if all links in said multiple link system have equal amounts of data queued for transmission,

selecting each link in sequence as the link on which to transmit data.

25. A multiple stage system for processing data stream on a multiple link system, said system comprising:

- a first stage for receiving data frames and extracting and storing payloads from said data frames;
- a second stage for rebuilding fragments from said payloads of said data frames; and

- a third stage for resequencing said fragments for eventual retransmission to a high speed data link,

wherein

each of said fragments forms a portion of a data packet, each of said fragments having a sequence number denoting a proper placement of said fragment in said reassembly and each of said fragments having a reassembly session number denoting a home packet of which said fragment is a part.

26. A system as in claim 25 wherein said first stage comprises:

- a data frame engine for extracting a payload from said data frames;

- a partial packet processor for storing payloads extracted from said data frames;

- an internal memory bank controlled by said processor such that said payloads are stored in said memory bank,

wherein

a specific memory location in said memory bank is allocated for each channel such that payloads from a specific channel are stored in said memory location.

27. A system as in claim 25 wherein said second stage comprises:

- a memory bank for storing data chunks, each data chunk being a portion of a fragment;
- a linked list pointer table, said pointer table having a record of which specific memory locations in said memory bank are used by specific data chunks wherein each data chunk relating to a specific fragment is stored in a specific memory location, an address of such specific memory location being placed in said pointer table and linked to addresses of previously stored data chunks.

28. A system as in claim 25 wherein said third stage comprises:

- at least one pointer array for each resequencing operation for storing head pointers, each head pointer representing a received fragment, each head pointer being associated with a slot in one of said at least one pointer array based on the sequence number of a fragment represented by said head pointer; and
- a lookup engine for determining if a fragment represented by a head pointer has a reassembly session number matching a resequencing operation being performed, wherein if a fragment has a reassembly session number matching a resequencing operation being performed, its head pointer

is placed in a pointer array for said resequencing operation.

29. A method as in claim 14 wherein said RAM bank is a synchronous dynamic random access memory (SRAM) bank.